UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/803,514	03/08/2001	Sridhar Obilisetty	026507-000300US	6110

20350 7590 08/25/2008

TOWNSEND AND TOWNSEND AND CREW, LLP
TWO EMBARCADERO CENTER
EIGHTH FLOOR
SAN FRANCISCO, CA 94111-3834

EXAMINER
VU, TUAN A

ART UNIT	PAPER NUMBER
2193	

MAIL DATE	DELIVERY MODE
08/25/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	09/803,514	OBILISETTY, SRIDHAR	
	Examiner	Art Unit	
	TUAN A. VU	2193	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *8/08/08*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-6,8-25,37 and 38* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-6,8-25,37 and 38* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. This action is responsive to the Applicant's response filed 8/08/2008.

As indicated in Applicant's response, claims 1, 13, 17, 25 have been amended, and claims 37-38 added. Claims 1-6, 8-25, 37-38 are pending in the office action.

Claim Rejections - 35 USC § 112

2. The following is a quotation of the first paragraph of 35 U.S.C. 112:

> The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

3. Claims 1-6, 8-25, 37-38 are rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the enablement requirement. The claim(s) contains subject matter which was not described in the specification in such a way as to enable one skilled in the art to which it pertains, or with which it is most nearly connected, to make and/or use the invention.

Specifically, claims 1, 13, 25 recite 'executing said application on said runtime environment independent from said program and asynchronously with respect to said compiling step'. The asynchronous mode of data transfer (ATM) or paradigm of communication with handling of asynchronous requests in client/server context as depicted in the Specifications was a well-known NW protocol, thus has no remote relation with the context recited in the above language including a client's compiling step and client's runtime environment's executing the application derived from said compiling. The server accepting simultaneous requests arriving in a asynchronous transfer protocol, in terms of non dependency between client instant state and server state as disclosed (Specifications: 2nd para - pg. 7; ATM, top pg. 16; pg. 19 bottom to pg. 10 top) has no remote linkage with an application being executed by a client exactly as required

from the claim language – execution of the application previously generated from said client device's resident program ('automatically compiling …'), the very application that when retrieved from storage and executed would be *asynchronous* with the very compiler's compiling step (said compiling step). That is, the 'executing asynchronously with respect to said compiling step' is not disclosed any where as to enable one of ordinary skill in the art to construe how the very *compiling step* (emphasis added on *said compiling step*) is being asynchronous (or synchronous) with a runtime execution instance of the application being compiled at a client device, then previously stored therein; and one of ordinary skill in the art would find highly unreasonable for a mechanism implementing synchronization (or non-synchronization) to be even applied for interrelating a compiler step to the event of executing said stored code. Lacking clear embodiments or utilities that explicitly describe one synchronizing mechanism for a compiling step to being synched or un-synched with the execution instance of an application derived from the 'automatically compiling' step, the inventor is deemed not in possession of this 'asynchronously' execution limitation at the time the invention was made. This limitation is given no weight and will be treated broadly as 'execution' of application subsequent to its being compiled.

Claims 2-6, 8-12, 14-24, 37-38 do not cure to the lack of description as raised above, hence are equally rejected for failing to comply with the enablement requirement.

Claims 37-38 recite 'removing said program from said computer ... executing said application … subsequent to the removing step'. This limitation is not provided with proper description in the Specifications. The nature of independency between client application flow and server service handling flow as depicted in the 'asynchronous' aspect of either environment

(see Specifications: pg. 20) only describes the non-blocking characteristic of application execution between client and server in a standard NW protocol; but nowhere is there mention of the act of actually **removing** the client's resident program (the one installed at the client and responsible for assembling the application) then **subsequent to such removing step**, executing the application compiled by this resident program in the client machine. The claims for not providing compliant support will be rejected as failing to comply with the enablement requirement. The limitation will be treated as though the execution is subsequent to a compilation step.

Claim Rejections - 35 USC § 103

4. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

5. Claims 1-6, 8-25, 37-38 rejected under 35 U.S.C. 103(a) as being unpatentable over Bloch et al, USPN: 2002/0129129 (hereinafter Bloch) and further in view of Murray et al, USPN: 6,874,143 (hereinafter Murray).

As per claim 1, Bloch discloses a method for implementing an application on a client computer system, said method comprising steps of:

a) receiving at said client computer system a plurality of text files (e.g. step 70-72, Fig. 5; para 0081-0082, pg. 9) wherein each of said text files defines a component of said application;

b) executing a program resident on said client computer system (*AVM 221* - Fig. 2; para 0068-0069, pg. 8; *GUI frame ... waits for the user* - para 0057-0058, pg. 6 – Note: installed and

iniatialized AVM -- para 0037, pg. 4; para 0050, pg. 5; Fig. 2-3 – by means of text script files and providing GUI rendered components for user events to take place reads on executing program resident on client), wherein said program comprises instructions for:

compiling automatically using a combination of versions of said text files to create said application (e.g. Fig. 2, 5; para 0044, pg. 4) wherein said application is stored on said client computer in a compiled form (para 00100, pg. 11; para 0102, pg. 11; para 0103, pg. 11; para 0105, pg. 12 - Note: executing a script with underlying invoking of procedures reads on compiled form being stored), and

executing said application in a runtime environment (e.g. *Client Tasks, Host tasks* – para 0086-87, pg. 10; *once ... initialized ... user interact ...visible frame ... next user* interaction … *System handler 315 to exit* - para 00100, pg. 11; *message box ... tasks ... unique to the operating system* – para 0102, pg. 11; *remote procedures* – para 0103, pg. 11; *Database handler ... user to access data ... Database* – para 0105, pg. 12; *user interface ... accessing any database ... to test new software* – para 0109; para 0062-0063, pg. 7) asynchronously with (Note: *asynchronously* treated as subsequent to – see USC 112 Rejection) respect to said compiling step.

Bloch does not explicitly disclose (1) checking automatically and periodically for updated versions of said text files; (2) receiving automatically any updated versions of said text files in response to said program checking for said updated versions when said updated versions are available; then (3) compiling periodically using said combination of said updated versions. Bloch discloses upgrades and fixes and possibility to make use of most recent versions (*upgrades, fixes* - pg. 3, para 0032; *most recent ...version* - pg. 12, para 0107) and addresses the urge for providing latest set of files in accordance to appropriate version of script file or virtual

machine files with regard to version (e.g. *receive upgrades as soon as possible* -para 0051-0053, pg. 5-6); hence has taught checking of files and their latest upgrade. Based on XML header or HTTP text files and browser (Fig. 5; para 0030, pg. 3), it is noted that a file version being included in the header of XML or HTTP files is implicitly disclosed owing to known browser/markup language technologies at the time the invention was made. **Murray**, in a similar framework to develop browser applications with the versions of XML or browser files, discloses a delivery of extension files (e.g. Fig. 15-16), with developer executing a client-side developing tool in contact with server environment to periodically check for download of latest upgrades for XML-implemented files (e.g. Fig. 14; *periodically polling* - col. 18 line 40 to col. 19 line 17). In light of the desirability of updating browser files to meet the appropriate virtual machine or execution environment version as mentioned above along with the implicitly automated version/format compliance of files processed by a browser engine, it would have been obvious for one of ordinary skill in the art at the time the invention was made to enhance Bloch's desire for proper version of files (para 0053, pg. 6) so that using the browser engine utilizes an automatic upgrade-checking mechanism via **periodic** inquiry or polling as taught in Murray in order for the latest upgraded XML or browser files to be retrieved for compilation (in a corresponding **periodic** scheme) as endeavored in Bloch's use of XML files, whereby extending/developing client or developer's applications. One would be motivated to do so because this periodic remote check/upgrade would enable the executing environment to be provided with the appropriate files according to the version expected for such environment as endeavored by both Bloch's approach and Murray's browser extension tool from above.

Nor does Bloch explicitly disclose stored compiled form so as to be executable independent from said program in a runtime environment independent from said program. Executing the script using the AVM by Bloch not only entails one instance of automatic script deployment for one client's AVM instance including possibility for the same user to reuse a previously compiled code (see *deploy application logic ... the application remains current with the developer's version* - para 0037, pg. 4), but also allowing reuse by other client's executing environment using a persisted and reusable code for distributed users; i.e. storing and persisting of applications in a distributed paradigm for use on specific platforms or families of similar machines in order to alleviate regeneration (e.g. *deploying and maintaining client in a distributed environment*, para 0105, pg. 12; *downloadable executable that installs quickly*, para 0106, pg. 12; *one set of software to be installed on all platforms* – para 0107, pg. 12; *over a distributed network to a variety of platforms* – para 0108, pg. 12) with possibility that one browser-rendering application derived from one set of neutral XML can be reutilized for variety of devices of similar GUI types (para 0096, pg. 11). It would have been obvious for one skill in the art at the time the invention was made to implement the application being generated by one AVM so that application can be persisted for reuse by the original user and/or on multiple users' environment having similar (or belong to a family of) platforms as taught above, because this would enable execution of reusable code (e.g. by same developer or multiple users via downloading code in their respective environment) **independent of the original compilation context**, enabling efficient reuse of resources in one's computer, and efficient distribution of resources executable to multiple users of similar platforms, expediting usage of such persisted

application, which is endeavored by Bloch, in terms of minimizing costs and quick and upgraded provisioning (see Bloch: para 0107- 0108 pg. 12)

As per claim 2, Bloch discloses XML format (Fig. 1, 2, 4,5).

As per claim 3, Bloch discloses server central source for managing and distributing applications or modifications for applications (e.g. *upgrades, fixes* - pg. 3, para 0032; pg. 5, para 0045-0046; pg. 12, para 0109).

As per claim 4, refer to claim 3 and Bloch's Fig. 1, 2, 4,5.

As per claim 5, Bloch discloses executing an application, sending a request and executing the application in parallel while waiting for response from the request (e.g. … *reports to the Application Handler 302*, … *periodically updates* -- pg. 9, para 0080 – 0082 – Note: resolving a URL with data retrieval while leaving the GUI window on for being updated on tree events changes and notified of download status is equivalent to executing application while waiting for remote response)

As per claim 6, Bloch discloses connectionless application execution (e.g. pg. 8, para 0069; pg. 12, para 0108)

As per claim 8, Bloch discloses modifying application by using a newer text files replacing older files (*upgrades, fixes* - pg. 3, para 0032; *most recent ...version* - pg. 12, para 0107).

As per claim 9, Bloch discloses graphical user interface (e.g. Fig. 6).

As per claim 10, Bloch discloses application being communication preferences for database invocation (e.g. pg. 7, para 0063; Preference Handler 303 - Fig. 4)

As per claim 11, Bloch discloses data management application (e.g. step *52* – Fig. 5; Manager 301 -Fig. 4 - Note: downloading files to assemble manager module reads on application being a management application).

As per claim 12, Bloch discloses component being part of logic of application (pg. 1, para 0012; pg. 4, para 0037).

As per claim 13, Bloch discloses a computer system comprising: a bus; a computer-readable memory unit coupled to said bus; and a processor coupled to said bus, said processor for executing a method for implementing an application comprising:

a)receiving at said computer system a plurality of text files wherein each of said text files defines a component of said application (refer to claim 1);

b)executing a program resident on said computer system (refer to claim 1), wherein said program comprises instructions for:

compiling automatically using a combination of versions of said text files to create said application (refer to claim 1), wherein said application is stored on said computer system; and executing said application on said runtime environment, asynchronously with respect to said compiling step (refer to claim 1).

Bloch does not explicitly disclose (1) checking automatically and periodically for updated versions of said text files; (2) receiving automatically any updated versions of said text files in response to said program checking for said updated versions when said updated versions are available; (3) compiling periodically using said combination of said updated versions. But these limitations have been addressed in claim 1.

Nor does Bloch explicitly disclose created application being stored compiled form so as to be executable independent from said program in a runtime environment independent from said program. But this limitation has been rendered obvious as set forth in claim 1.

As per claims 14-18, 20-24, these claims correspond to claims 2-6, 8-12 respectively, hence are rejected with the corresponding rejections as set forth therein, respectively.

As per claim 19, Bloch discloses text files particular to client system (e.g. pg. 4, para 0037; pg. 5, para 0047, 0050)

As per claim 25, Bloch discloses a computer-usable medium having computer-readable program code embodied therein for causing a computer system to perform a method comprising code for:

a)installing on said computer system a plurality of text files (refer to claim 1) wherein each of said text files defines a component of said application;

b) installing a program on said computer system (para 0057, pg 6), wherein said program comprises instructions for:

compiling automatically using a combination of versions of said text files to create said application program (refer to claim 1), wherein said application is stored on said computer system in a compiled form (refer to claim 1), and executing said application asynchronously with respect to said compiling step (refer to claim 1).

Bloch does not explicitly disclose (1) checking automatically and periodically for updated versions of said text files; (2) receiving automatically any updated versions of said text files in response to said program checking for said updated versions when said updated versions

are available; (3) compiling automatically and periodically using combination of said updated versions of said text files. But these limitations have been addressed in claim 1.

Nor does Bloch explicitly disclose created application being stored in compiled form so as to be executable independent from said program in a runtime environment independent from said program. But this limitation has been rendered obvious as set forth in claim 1.

As per claims 37-38, Bloch discloses (by virtue of the **USC 112 1st rejection**) executing said application on said client computer system (e.g. *Client Tasks, Host tasks* – para 0086-87, pg. 10; *once ... initialized ... user interact ...visible frame ... next user* interaction … *System handler 315 to exit* - para 00100, pg. 11; *message box ... tasks ... unique to the operating system* – para 0102, pg. 11; *remote procedures* – para 0103, pg. 11; *Database handler ... user to access data ... Database* – para 0105, pg. 12; *user interface ... accessing any database ... to test new software* – para 0109; para 0062-0063, pg. 7) subsequent to the removing step (Note: removing step treated as a compilation stage).

Response to Arguments

6. Applicant's arguments filed 8/08/2008 have been fully considered but they are not persuasive or MOOT in view of the new grounds of rejection. Following are the Examiner's observation in regard thereto.

USC 103(a) Rejection:

(A) Applicant has submitted that as amended, the limitation 'compiled form so as to be executable independent from said program in a runtime … independent from said program' in claims 1 ,13 and 25 is not disclosed in Bloch (Appl. Rmrks pg. 9); e.g. because, using an agent to compile an application, the program can still be executed once the agent is removed. The claim

does not nearly implicate removing a agent; nor does the limitation belong to the subject matter being addressed in the Previous Office Action. The argument is considered moot in view of the Amendments.

(B) Applicant has submitted that many portions in Bloch support the fact that once the AVM is removed, the application cannot be executable as claimed, because Bloch's script executes within a virtual application assembling of XML files to support some GUI rendering specific to a platform, the assembled application truly dependent on the AVM, execution ends with the closing of the AVM (Appl. Rmrks pg. 10-12). The Amendments to the claims has triggered adjusted grounds of rejection; and the above remarks are deemed premature and largely misplaced.

In all, the newly amended claims stand rejected as set forth in the Office Action.

Conclusion

7. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Tuan A Vu whose telephone number is (571) 272-3735. The examiner can normally be reached on 8AM-4:30PM/Mon-Fri.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Lewis Bullock can be reached on (571)272-3759.

The fax phone number for the organization where this application or proceeding is assigned is (571) 273-3735 (for non-official correspondence - please consult Examiner before using) or 571-273-8300 (for official correspondence) or redirected to customer service at 571-272-3609.

Any inquiry of a general nature or relating to the status of this application should be directed to the TC 2100 Group receptionist: 571-272-2100.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

/Tuan A Vu/

Primary Examiner, Art Unit 2193

August 23, 2008